On Product Comparison Matrices and Variability Models from a Product Comparison/Configuration Perspective

Nicolas Sannier, Guillaume Bécan, Sana Ben Nasr, Benoit Baudry

On Product Comparison Matrices and Variability Models from a Product Comparison/Configuration Perspective

Nicolas Sannier[*] — Guillaume Bécan[*] — Sana Ben Nasr[*] — Benoit Baudry[*]

[*] *Inria Rennes Bretagne Atlantique / IRISA - Université de Rennes 1*
EPI Triskell - Campus de Beaulieu - 35042 Rennes Cedex
firstname.lastname@inria.fr

ABSTRACT. *Comparators and configurators have now become common in our daily activities and are usually based on Product Comparison Matrices (PCMs) to present and compare features. Based on a previous analysis of 300+ PCMs from Wikipedia, we identify the limits of existing comparators, configurators and PCMs. Variability Models (VMs) have been extensively used through the last 20 years to provide a synthetic and formal way to represent a product line. As a consequence, using VMs instead of PCMs could tackle these limits and improve comparison and configuration activities. In this paper, we present 5 research questions that focus on using VMs to represent PCMs and their applications for comparators and configurators.*

RÉSUMÉ. *Les comparateurs et configurateurs de produits sont devenus des objets du quotidien et sont souvent représentés sous la forme de tableaux. L'analyse de 300+ tableaux de comparaison issus de Wikipedia a montré les limites de ceux-ci, en plus de celles des comparateurs et configurateurs. Les modèles de variabilité (MV) proposent une vue formelle et synthétique d'une ligne de produits. La formalisation de MVs à partir de matrices de comparaison permettrait d'aller au delà de ces limites et de proposer des outils de comparaison et de configuration plus avancés. Dans cet article, nous proposons 5 questions de recherche autour de l'utilisation de MVs pour la formalisation de matrices de comparaison et leur utilisation dans le cadre de comparateurs et configurateurs.*

KEYWORDS: *Products Comparison Matrices, Products Configurators, Variability Information, Feature Models.*

MOTS-CLÉS : *Tableaux de comparaisons de produits, Configurateurs, Informations de variabilité, Modèles de features.*

1. Introduction and Motivation

Products comparators and configurators have now become common in our daily activities. Whenever one desires to compare prices, buy a new camera, a computer, a smartphone, a car or even a shirt, he or she can observe or compare or be assisted during the process that will eventually lead to the concrete selection and buying of a product.

If we consider commercial comparison pages for different domains, we observe that these comparison are mostly provided into a tabular way and a product $\times$ feature perspective. These Product Comparison Matrices (PCMs) provide a simple and convenient way to express properties on products and compare them to several different others from the same *family*. They are provided by open initiatives like Wikipedia or consumers organizations. It allows companies to present and advertise on the different *facets* of their *product series*. PCMs provide a global view on several different competing products, showing the presence, absence, limitations of a facet, expressing *commonality and variability* between products under comparison.

From a general perspective, we can observe the different limitations for comparators and PCMs:

– comparisons are limited to a maximum number of products,

– compared "attributes" are presented as a list with a predefined and fixed order,

– the more attributed there is, the less the table is readable and understandable.

Some comparison tools also propose an evaluation and ranking service, based on the information they can exploit, but they are not really accurate. This means that the tasks of filling, interpreting and analyzing the comparison information is almost human-based from both the developer and the user perspective.

By the same way, product configurators propose to build more "personalized" products accordingly to personal choices over some of the product features. Current configurators lack from flexible configuration scenarios, as configurations are almost done within a limited and fixed set of steps. They do not also take into account all the product features in their complete scope (Perrouin *et al.,* 2008). As a consequence one may question how deep is the personalization process while configuring a product.

On the one hand, these limitations for comparators and configurators set exciting questions and challenges for the product line community in terms of services expressiveness and tooling in order to propose flexible, attractive, assisted comparison and configuration services. On the other hand, the product line community has a long past experience with variability modeling and software product line engineering.

Variability Models (VMs), like decision models or feature models, share similar goal than PCMs and provide a very synthetic and visual way to describe all possible products (also called *configurations*) in a given domain. VMs are an alternative formalism to overcome the previous identified limitations of PCMs. VMs can be employed to formulate in a more formal way the meanings of PCMs. VMs and their formal semantics enable automated reasoning: VMs come with state-of-the-art satisfiability techniques and solvers that can be used for performing assisted configurations. VMs offer an explicit and compact view of the variability and logical relationships between features.

Choosing a configuration from a VM then means selecting and deselecting features with respect to the FMs constraints while a configuration corresponds to a given product that matches the feature selection. Our intuition is that leveraging the 20+ years experience (Benavides *et al.,* 2010) on feature and variability modeling can help to tackle these comparison and configuration issues. Expected advantages and benefits from variability modeling approaches for that question are summed up as:

– Formalization of the variability information
– 20+ years of analyses and tool-supported work
– Independence from the feature selection
– Synthetic and exhaustive representation formalism

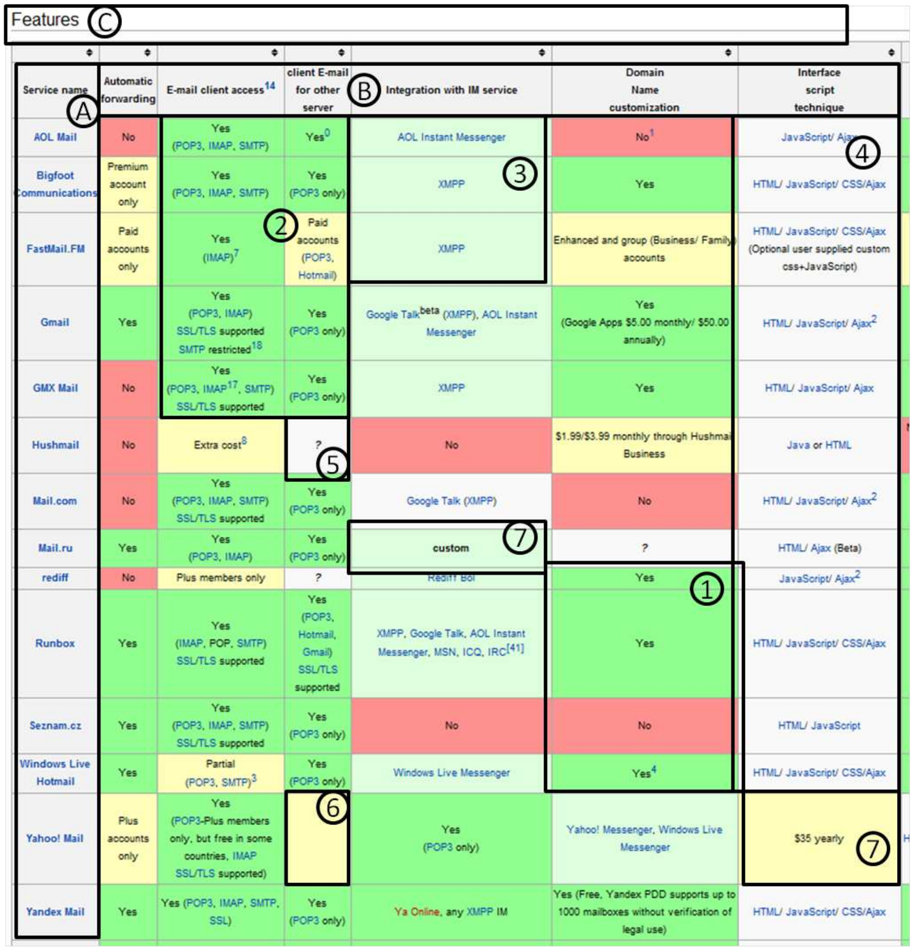
We previously evaluated PCMs contents over 300+ PCMs in Wikipedia (Sannier *et al.,* 2013). We proposed a set of variability information types that goes beyond the traditional boolean perspective of PCMs and that have to be taken into account if one want to fully exploit PCMs.

The global research questions we are interested in are as follows. Is a variability model a sound and consistent formalism if we want to provide some support over these PCMs ? Provided such a variability model, can we propose methodologies and tools for comparison and configuration purposes? With these two questions and based on this PCM variability information, we propose to address, in this paper, some research directions and challenges toward PCM formalization and tooling in order to propose more flexible and richer product comparators and configurators.

The remainder of the paper is as follows. Section 2 presents the anatomy of a Wikipedia PCM, the taxonomy of variability information that we defined in (Sannier *et al.,* 2013). Section 3 discusses the step between PCMs and their formalization within variability models and calls for more research effort. Section 4 concludes the paper.

2. PCM Variability Information

In this section, we present the main results on our analysis about variability information from Wikipedia Product Comparison Matrices, and present meaningful information on their contents. We first present an example based on one wikipedia PCM and present our variability information patterns. We then evaluate their proportion through two evaluations. The first evaluation is based on a qualitative analysis on 50 randomly selected Wikipedia PCMs. The second is an automatic analysis of a corpus of 300+ PCMs.



Features

Service name	Automatic forwarding	E-mail client access[14]	client E-mail for other server	Integration with IM service	Domain Name customization	Interface script technique
AOL Mail	No	Yes (POP3, IMAP, SMTP)	Yes[0]	AOL Instant Messenger	No[1]	JavaScript/ Ajax
Bigfoot Communications	Premium account only	Yes (POP3, IMAP, SMTP)	Yes (POP3 only)	XMPP	Yes	HTML/ JavaScript/ CSS/Ajax
FastMail.FM	Paid accounts only	Yes (IMAP)[7]	Paid accounts (POP3, Hotmail)	XMPP	Enhanced and group (Business/ Family) accounts	HTML/ JavaScript/ CSS/Ajax (Optional user supplied custom css+JavaScript)
Gmail	Yes	Yes (POP3, IMAP) SSL/TLS supported SMTP restricted[18]	Yes (POP3 only)	Google Talk[beta] (XMPP), AOL Instant Messenger	Yes (Google Apps $5.00 monthly/ $50.00 annually)	HTML/ JavaScript/ Ajax[2]
GMX Mail	No	Yes (POP3, IMAP[17], SMTP) SSL/TLS supported	Yes (POP3 only)	XMPP	Yes	HTML/ JavaScript/ Ajax
Hushmail	No	Extra cost[8]	?	No	$1.99/$3.99 monthly through Hushmail Business	Java or HTML
Mail.com	No	Yes (POP3, IMAP, SMTP) SSL/TLS supported	Yes (POP3 only)	Google Talk (XMPP)	No	HTML/ JavaScript/ Ajax[2]
Mail.ru	Yes	Yes (POP3, IMAP)	Yes (POP3 only)	custom	?	HTML/ Ajax (Beta)
rediff	No	Plus members only	?	Rediff Bol	Yes	JavaScript/ Ajax[2]
Runbox	Yes	Yes (IMAP, POP, SMTP) SSL/TLS supported	Yes (POP3, Hotmail, Gmail) SSL/TLS supported	XMPP, Google Talk, AOL Instant Messenger, MSN, ICQ, IRC[41]	Yes	HTML/ JavaScript/ CSS/Ajax
Seznam.cz	Yes	Yes (POP3, IMAP, SMTP) SSL/TLS supported	Yes (POP3 only)	No	No	HTML/ JavaScript
Windows Live Hotmail	Yes	Partial (POP3, SMTP)[3]	Yes (POP3 only)	Windows Live Messenger	Yes[4]	HTML/ JavaScript/ CSS/Ajax
Yahoo! Mail	Plus accounts only	Yes (POP3-Plus members only, but free in some countries, IMAP SSL/TLS supported)		Yes (POP3 only)	Yahoo! Messenger, Windows Live Messenger	$35 yearly
Yandex Mail	Yes	Yes (POP3, IMAP, SMTP, SSL)	Yes (POP3 only)	Ya Online, any XMPP IM	Yes (Free, Yandex PDD supports up to 1000 mailboxes without verification of legal use)	HTML/ JavaScript/ CSS/Ajax

Figure 1 – A family of online emails products

2.1. *Anatomy of a Wikipedia PCM*

As for illustrative purpose, we analyze a PCM about webmail providers mined in Wikipedia [1] and present a sample of the PCM in figure 1. This PCM compares 15 different products (A in the figure) against 12 different criteria (B in the figure). This Wikipedia page also proposes different comparison perspectives (C) and, consequently, several PCMs related to these perspectives. However, our example focuses on the PCM of figure 1, which includes 180 different cells to analyze.

1. Available online at `http://en.wikipedia.org/wiki/Comparison_of_webmail_providers`, last access 10th may 2013

The first observation we make is related to the different comparison criteria, found as headers of the PCM. A PCM is composed of a list of heterogeneous criteria with different levels of precision and flexibility. Consequently, products values regarding these criteria can be a various kind such as:

– ① **Boolean yes/no values**. This kind of variability deals with the straight, non ambiguous, presence or absence of the comparison criteria. We observe that couples of tokens like "yes/no", "true/false", etc. are potential candidates for this kind of variability information.

– ② **Constrained/Partial/ambiguous yes/no values**. This kind of cells has to be interpreted as: "the criterion is satisfied under the condition of, with the following limitation, etc"."Only", "if", "through", can be candidate words to recognize this kind of value. The token "partial" is the most significant evidence of the presence of the value type. One can also see a "yes" with a footnote or followed by one or several elements that express a condition or limitation.

– ③ **Single-value**. This kind of information has to be interpreted as: "the criterion is satisfied using this element". It forms a unique way to satisfy the criteria. The purpose of this information is not to know whether or not the criterion is satisfied but how.

– ④ **Multi-values**. This kind of information has to be interpreted as: "the criterion is satisfied using these elements". It forms a set of elements that contributes to satisfy the criterion. It should be noted that there is no homogeneity, within the same matrix, in the way of expressing such enumerations.

– ⑤ **Unknown value**. One does not know if the criterion is satisfied. Cells are generally filled with "?", "unknown". This information is rather hard to manage. It cannot be fully interpreted as a boolean "no" answer, as it can prevent the product from being selected, despite the domain reality that is unknown.

– ⑥ **Empty cell**. This information is hard to interpret, i.e., whether it should be analyzed as a strong boolean "no" and accordingly assessed as the absence of the feature or should this be analyzed as an unknown answer ?

– ⑦ **Inconsistent value**. The provided value is partial, ambiguous or lightly related to the analyzed criterion. For instance, in Figure 1, it is mentioned that "Yahoo! Mail" has a "$35 yearly" interface, whereas all other products mention the underlying technology of their interface.

– ⑧ **Extra Information**. The provided cell value offers additional information such as latest dates or versions. Though not present in Fig 1, this pattern exists.

It should be noted that the eight information types defined above are not necessarily expressed in a regular way for a given criteria/header. Specifically, a same header can refer to a specific value for one product, be unknown for another one, conditionally active in another case, etc. An example is given for the header feature "Client access for email Server" (see Fig. 1).

Table 1 – Value types frequencies for 50 Wikipedia PCMs

	①	②	③	④	⑤	⑥	⑦	⑧
Total	47,29%	3,71%	22,75%	4,37%	10,86%	4,83%	0,55%	5,64%

2.2. *A Qualitative Analysis of 50 Wikipedia PCMs*

We want to further confirm our intuition over PCM contents. For this purpose, we analyzed a randomly selected sample of 50 Wikipedia's PCMs.

Table 1 provides a summary of our analysis of the 50 Wikipedia pages, the number of tables, cells, and values frequency [2]. These 50 pages contained 165 tables and about 29500 different cells. The 50 pages mainly deal with computer systems, architectures, programming at various levels but also include topics like linguistic, mechanics, politics, defense, among others.

Concerning "uncertainty", information that is not a straightforward variability information (②, ⑤, ⑥, ⑦, and ⑧), it represents a mean of 25.6%. It represents a significant number of cells that cannot stand as-is in a FM. On the other hand, around 75% of PCMs content is rather direct information and allow a direct mapping to FMs.

2.3. *A Quantitative Analysis of 300+ Wikipedia PCMs*

To gain further statistical evidence about the frequency of the eight patterns, we implemented an automated extraction process for operating over 300+ Wikipedia pages. We used the state of the art parser *Sweble* (Dohrn *et al.,* 2011) to process the source of each Wikipedia page. In addition, we implemented automated techniques to recognize the pattern of a cell value, following the observations of the qualitative study. We do not seek to automatically detect patterns ⑦ and ⑧ since they are mainly based on human perception.

In total, we analyzed 31097 products and 225024 cell values. The results are reported in Table 2.

2.4. *Discussion on PCMs Variability Information*

We now compare the results with those previously obtained in the qualitative study. The frequency of Boolean values has slightly increased (49.4 *versus* 47.3) and still important, confirming the importance of the pattern ① . Similarly, the frequency of

2. More detailed information for each page is available online at `http://tinyurl.com/WikipediaPCM`

Table 2 – Value types frequencies for 300+ Wikipedia PCMs

	①	②	③	④	⑤	⑥
amount	111309	1788	45903	33922	16823	15279
%	49.4	0.8	20.4	15.1	7.5	6.8

single values (pattern ③) remains important (slight decrease with 20.4 *versus* 22.75). The frequency of multi-values ④ has increased to a large proportion (15.1 *versus* 4.37). We can hypotesize that part of the values can actually belong to pattern ⑦ or ⑧ (two patterns we do not detect and that are usually constituted of multiple values). The frequency of pattern ② has decreased significantly (0.8 *versus* 3.71) but still constitutes a minor pattern.

The most important result is that we confirm patterns ① and ③ are by far the most widely used, constituting almost 75% of the content of PCMs.

3. Research Directions

In this section, we address two different challenges that were defined previously. We first address the variability model generation itself. We then propose a set of research questions concerning the tooling capability and services over these variability models within a comparator and configurator perspective.

We recall PCMs limitations and we add them to those we highlighted regarding comparison and configuring concerns.

– Comparisons are limited to a maximum number of products (or all of them in Wikipedia).

– Compared "attributes" are presented as a list with a predefined and fixed order.

– Cells in a PCM lack of formalization as they can provide a simple yes/no information describing the presence/absence of a feature, numerical information,unknown values or even implicit empty cells. All these values have to be interpreted w.r.t. variability.

– The size and complexity of data can be very important, up to hundreds of products and hundreds of features. Consequently, it can be hard to understand and exploit a PCM of such size. Users must review numerous cells of the PCM to gather the required information regarding his/her requirements. In practice, the more criteria or products there are, the harder the PCM is to read, the higher the probability is to miss important information and the less a user can make an effective choice.

– Configurations are almost done within a limited and fixed set of steps and features.

Based on our previous analyses, we have quantitative evidence upon PCMs limitations. In particular, we observe that about 75% of PCMs are VM-compatible. On the other hand, there is another 25% part that remains difficult to interpret and analyze.

This lack of formalization hinders the construction of tools and prevents forward efficient and systematic analyses on PCMs. It then becomes hard to propose efficient configurators that could be based on this precious information. Offering tools and/or good practice guidelines for internet users who are filling these matrices would be a first step towards formalization. It could reduce the part of uncertainty and highlight patterns in PCMs, thus indirectly bridging the gap between PCMs and VMs. Now, we present 5 research questions that focus on using VMs for representing PCMs and their applications in comparators and configurators.

3.1. *Toward Rich reverse-engineering of Variability Models with PCMs*

RQ$_1$: How to formalize and deal with these 25% remaining information? What does this information means from a variability point of view? A quarter of the information represent non boolean values such as numerical values, enumerations or uncertainty. Boolean VMs fail to formalize such types of values. Thus, another formalism is required. Some papers propose extensions of FMs that could be candidates for formalizing PCMs.

Benavides *et al.* propose an extension including attributes (Benavides *et al.,* 2005). Each attribute define a value contained in the attribute's domain (*e.g.* integer, real, enumeration, boolean). For example, we can define the weight of a camera lens by adding an attribute *weight* in the feature *Lens*. These attributes aim at modelling extra-functional features (*e.g.* a lens must weight less than 1kg). Cordy *et al.* extend FMs with attributes and multi-features (Cordy *et al.,* 2013). Multi-features allow products to have several instances of a same feature. For example, we could easily model that a computer has several hard drives. Such attributes and multi-features could formalize the non boolean values of PCMs. However, these extended FMs do not offer constructs for defining uncertainty. Czarnecki *et al.* propose a different extension of FMs called probabilistic FMs (Czarnecki *et al.,* 2008). Probabilistic FMs allow to define soft constraints between features. For example, 80% of the cameras of the product line are bought with a 50mm lens.

In addition to VMs, there exist other methods such as formal concept analysis (Loesch *et al.,* 2007, Ryssel *et al.,* 2011) that could represent the variability contained in PCMs or help us obtain VMs but no work target variability of PCMs. Therefore, we need to define the possible semantics of PCMs and try to map this semantics to attributed FMs, probabilistic FMs or another formalism. Once an adequate formalism is found, we could offer tools or guidelines to help users in defining PCMs with a precise semantics.

RQ$_2$: How to efficiently synthesize consistent and meaningful VMs from PCMs? As stated before, the size and complexity of PCMs can be very important.

As a result, the manual elaboration of a VM from such PCMs is time-consuming and error-prone. Automating the process presents 4 challenges.

First, the resulting VM must be consistent, *i.e.* representing the same product line as the input PCM. An inconsistent VM may expose the user to invalid configurations. Therefore, he or she may purchase products that simply do not exist. Second, the resulting VM shall present a meaningful hierarchy of features for its users (*e.g.* a meaningful feature diagram for FMs). Such hierarchy ease the understanding and the maintenance of the VM. It can also ease the development of comparators and configurators by presenting to developers a default yet meaningful scenario. Third, an automated synthesis must be able to correctly interpret the uncertainty in PCMs. If the PCM do not have a precise semantics, the user should disambiguate each cell that contains uncertainty. Finally, an automated synthesis should be efficient and scale on large PCMs.

Acher *et al.* propose an automated technique to reverse engineer boolean FMs directly from PCMs (Acher *et al.,* 2012). Other works focus on reverse engineering boolean FMs from different input artefacts (Czarnecki *et al.,* 2007, Andersen *et al.,* 2012, She *et al.,* 2011, Acher *et al.,* 2013, Haslinger *et al.,* 2011, Haslinger *et al.,* 2013, Ryssel *et al.,* 2011, Ziadi *et al.,* 2012, Weston *et al.,* 2009, Chen *et al.,* 2005, Al-Msie'deen *et al.,* 2013). For example, Andersen *et al.* developed an efficient technique for reverse engineering a consistent FM from a propositional formula (Andersen *et al.,* 2012). Ryssel et al. used formal concept analysis to extract FMs from incidence matrices which may be seen as PCMs. She *et al.* addressed the problem of synthesizing a meaningful FM by using user input and a domain-specific heuristic to rank parent candidates (She *et al.,* 2011). We recently developed a generic technique that also uses user input and heuristics for reverse engineering a FM from a propositional formula (Bécan *et al.,* 2013). However, all these techniques produce boolean FMs. Czanercki *et al.* proposed to reverse engineer probabilistic FMs but the resulting FM represents a superset of the product line configuration set (Czarnecki *et al.,* 2008). To our knowledge, no techniques for reverse engineering attributed FMs were proposed.

Therefore, we need an efficient automated technique for reverse engineering the VM identified in RQ_1. This technique shall allow the user to provide knowledge about the hierarchy of the VM and the semantics of the PCM's cells in order to obtain a consistent and meaningful VM.

3.2. *From Variability Models to a Unified Approach for User Guidance in Comparators and Configurators*

RQ_3: When does it become beneficial to use a VM instead of a matrix? PCMs are quite natural for describing product lines but as we stated before, they present some limits especially when their size increases. Using VMs would ease the maintenance and the understanding of PCMs. However, switching from PCMs to VMs is not straitghforward and requires substantial effort (see RQ_2). Defining criteria from which it is necessary to switch from PCMs to VMs, would help users in selecting the right formalism.

RQ_4: How to reason and analyze over VMs for comparison and configuration? In their literature review, Benavides et al. reported that reasoning and analyzing over FMs is mostly done with propositional logic based techniques such as SAT solvers or Binary Decision Diagrams (Benavides *et al.,* 2010). They also noted that extended FMs were not supported by these techniques in the papers they studied. However, constraint programming based techniques seemed more suited for analyzing attributed FMs as they allow to reason on both boolean and numerical values. Other solutions exists such as satisfiability modulo theories (SMT) solvers (Cordy *et al.,* 2013). Czarnecki *et al.* mentioned several techniques for using probabilistic FMs in a configuration process but they did not validate them yet (Czarnecki *et al.,* 2008).

Cordy *et al.* mentioned that using SMT solvers for attributed FMs significantly increase reasoning time (Cordy *et al.,* 2013). We can expect the same behaviour for other techniques that are not based on propositional logic. As PCMs can contain hundreds of products and features, the performance issue is a major concern. Users do not want comparators and configurators that take several minutes to take into account each choice.

Therefore, we need an efficient technique that handles all the types of values that we can encounter in a PCM (boolean, numerical, enumeration, probabilities, etc.).

RQ_5: How to build comparators and configurators from VMs? To address the limits of comparators and configurators identified in this paper, we need to address the following challenges. First, comparator and configurator systems shall make available all the pieces of informations contained in the VM. Thereby the user can check all his/her requirements. The challenge is to provide such amount of information without overwhelming the user. Second, these systems must include flexibility in the comparison and configuration process. Only constraints between features and marketing requirements should be enforced. Again, comparison or configuration systems should implement mecanisms to avoid overwhelming users with numerous possible scenarios. For example, such systems could use default values for mandatory features, multi-step configuration process (Khalil Abbasi *et al.,* 2013), recommender systems (Salinesi *et al.,* 2012, Dumitru *et al.,* 2011), graphical interface for finding optimum products (Murashkin *et al.,* 2013) or soft constraints expressed by the user (Czarnecki *et al.,* 2008). Finally, comparators and configurators should be intuitive and documented. Good practices such as explaining constraints and features, presenting examples, using standard graphical widgets and allowing to remove the last decision would help the user in his or her task (Khalil Abbasi *et al.,* 2013).

All these good practices and functionalities could be integrated in a framework that ease the development of comparators and configurators. Developers would only define the VM, the different constraints on the product line and the desired graphical interface.

4. Conclusion

In this paper, we identified several limits to current comparators, configurators and Product Comparison Matrices (PCMs). In particular, we studied 300+ PCMs from Wikipedia and showed that 25% of PCMs' cells contain uncertainty. This leads us to investigate how to better formalize PCMs through Variability Models (VMs).

VMs offer a synthetic and formal way to represent product lines. Such formalism could improve comparators and configurators and tackle their limits. Many works proposed techniques for reverse engineering, analyzing and exploiting boolean VMs. However, boolean VMs fail to formalize numerical values, enumerations and the uncertainty contained in PCMs. Few extensions such as attributed and probabilistic VMs were proposed to handle these types of information but there still exists a gap between PCMs and VMs.

We presented a set of reasearch questions that focus on bridging this gap by mapping the semantics of PCMs to an appropriate VM and developing techniques to reverse engineer this VM from one or several PCMs. We are also interested in defining criteria from which it is necessary to switch from PCMs to VMs in order to help users in choosing the right formalism. Finally, we presented challenges for building efficient, flexible and intuitive comparators and configurators.

5. References

Acher M., Cleve A., Perrouin G., Heymans P., Vanbeneden C., Collet P., Lahire P., « On extracting feature models from product descriptions », *VaMoS'12*, ACM, p. 45-54, 2012.

Acher M., Heymans P., Cleve A., Hainaut J.-L., Baudry B., « Support for Reverse Engineering and Maintaining Feature Models », *VaMoS'13*, ACM, Pisa, Italie, jan, 2013.

Al-Msie'deen R., Seriai A.-D., Huchard M., Urtado C., Vauttier S., « Mining features from the object-oriented source code of software variants by combining lexical and structural similarity », *IRI'13*, p. 586-593, 2013.

Andersen N., Czarnecki K., She S., Wasowski A., « Efficient synthesis of feature models », *SPLC'12*, ACM Press, p. 97-106, 2012.

Bécan G., Acher M., Baudry B., Ben Nasr S., Breathing Ontological Knowledge Into Feature Model Management, Technical report, Inria, 2013.

Benavides D., Ruiz-Cortés A., Trinidad P., « Automated Reasoning on Feature Models », *CAiSE 2005*, vol. 3520, p. 491-503, 2005.

Benavides D., Segura S., Ruiz-Cortes A., « Automated Analysis of Feature Models 20 years Later: a Literature Review », *Information Systems*, 2010.

Chen K., Zhang W., Zhao H., Mei H., « An approach to constructing feature models based on requirements clustering », *RE'05*, p. 31-40, 2005.

Cordy M., Schobbens P.-Y., Heymans P., Legay A., « Beyond boolean product-line model checking: dealing with feature attributes and multi-features », *ICSE 2013*, IEEE Press, p. 472-481, 2013.

Czarnecki K., She S., Wasowski A., « Sample Spaces and Feature Models: There and Back Again », *SPLC '08*, p. 22-31, 2008.

Czarnecki K., Wasowski A., « Feature Diagrams and Logics: There and Back Again », *SPLC'07*, IEEE, p. 23-34, 2007.

Dohrn H., Riehle D., « Design and implementation of the Sweble Wikitext parser: unlocking the structured data of Wikipedia », *WikiSym '11*, ACM, New York, NY, USA, p. 72-81, 2011.

Dumitru H., Gibiec M., Hariri N., Cleland-Huang J., Mobasher B., Castro-Herrera C., Mirakhorli M., « On-demand feature recommendations derived from mining public product descriptions », *ICSE '11*, ACM, New York, NY, USA, p. 181-190, 2011.

Haslinger E. N., Lopez-Herrejon R. E., Egyed A., « Reverse Engineering Feature Models from Programs' Feature Sets », *WCRE'11*, IEEE CS, p. 308-312, 2011.

Haslinger E. N., Lopez-Herrejon R. E., Egyed A., « On Extracting Feature Models from Sets of Valid Feature Combinations », *in* , V. Cortellessa, , D. Varró (eds), *FASE*, vol. 7793 of *Lecture Notes in Computer Science*, Springer, p. 53-67, 2013.

Khalil Abbasi E., Hubaux A., Acher M., Boucher Q., Heymans P., « The Anatomy of a Sales Configurator: An Empirical Study of 111 Cases », *in* , M. Norrie, , C. Salinesi (eds), *CAiSE'13*, Valencia, Espagne, jun, 2013.

Loesch F., Ploedereder E., « Restructuring Variability in Software Product Lines using Concept Analysis of Product Configurations », *CSMR '07: Proceedings of the 11th European Conference on Software Maintenance and Reengineering*, IEEE Computer Society, Washington, DC, USA, p. 159-170, 2007.

Murashkin A., Antkiewicz M., Rayside D., Czarnecki K., « Visualization and exploration of optimal variants in product line engineering », *SPLC'13*, 2013.

Perrouin G., Klein J., Guelfi N., Jézéquel J.-M., « Reconciling Automation and Flexibility in Product Derivation », *SPLC'08*, IEEE, p. 339-348, 2008.

Ryssel U., Ploennigs J., Kabitzsch K., « Extraction of feature models from formal contexts », *FOSD'11*, p. 1-8, 2011.

Salinesi C., Triki R., Mazo R., « Combining configuration and recommendation to define an interactive product line configuration approach », *CoRR*, 2012.

Sannier N., Acher M., Baudry B., « From Comparison Matrix to Variability Model: The Wikipedia Case Study », *ASE'13*, IEEE, p. 580-585, 2013.

She S., Lotufo R., Berger T., Wasowski A., Czarnecki K., « Reverse Engineering Feature Models », *ICSE'11*, ACM, p. 461-470, 2011.

Weston N., Chitchyan R., Rashid A., « A framework for constructing semantically composable feature models from natural language requirements », *SPLC'09*, ACM, p. 211-220, 2009.

Ziadi T., Frias L., da Silva M. A. A., Ziane M., « Feature Identification from the Source Code of Product Variants », *in* , T. Mens, , A. Cleve, , R. Ferenc (eds), *CSMR*, IEEE, p. 417-422, 2012.